Based on the forthcoming:

Blaauw, F.J., Chambaz, A., Van der Laan, M.J., (2017). Exploring the causal effects of activity on well-being using online targeted learning. *In preperation*.

Chapter 8

Exploring the causal effects of activity on well-being using online targeted learning

Being healthy is a state in which one has complete physical, mental, and social well-being, thus not merely the lack of diseases or infirmity (World Health Organization, 2014). Research in psychopathology epidemiology has mainly focused on exploring factors that cause *mental ill-being*, as opposed to well-being (Slade, 2010). Recently, researchers have shown a renewed interest in a hybrid approach, in which both the absence of mental illness and the presence of well-being play an important role (e.g., Keyes, 2007; Lee Duckworth, Steen, & Seligman, 2005; Slade, 2010; van der Krieke, Jeronimus, et al., 2016). Instead of focusing on reducing the illness, the focus is shifted towards the increase of well-being and positive affect. Another important shift in psychopathology research is the shift towards a more personalized, individualistic approach. Instead of determining results that hold on the group or population levels, researchers slowly shift to determining those results on the individual level. Arguably, the shift in focus will eventually allow for a more personalized medicine.

Many methods exist to assess which factors affect well-being based on large cohort studies. However, results from these studies provide information on the group level and are insufficient for inference at the individual level (Hamaker, 2012; Molenaar & Campbell, 2009). Indeed, by heterogeneity among people the results might be true on average, but might not actually hold for *any* individual (Lamiell, 1998). Finding the factors that influence well-being on the individual level has proved to be a challenge, and new techniques need to be devised to determine these individual differences in the factors aiding well-being.

In this chapter, we describe the theoretical background and implementation of a novel machine learning technique that can help us to answer questions of the form: *"How does well-being change over time for a specific individual, when intervening on general activity at the preceding times?"* To determine the performance of this technique,

we perform an initial simulation study and exploratory analysis on the causal effect of a specific factor, namely general activity, on the well-being of an individual.

Our analysis of the causal effect of general activity on the well-being of an individual relies on a specific data set from the large Dutch mental-health study HowNutsAreTheDutch (HND, see Chapter 3 for more details). It is a novel instantiation of the *targeted learning* methodology (Petersen & van der Laan, 2014; van der Laan & Rose, 2011) tailored to infer from a single time series (Benkeser, Ju, Lendle, & van der Laan, 2016; van der Laan & Rose, 2017). In addition to analyzing the HND data set, we illustrate how the tailored instantiation performs in a preliminary simulation study. Although our main focus is on finding results that hold for a specific individual, we also apply our method to the whole group.

8.1 Quick Historical Overview of the Targeted Learning Methodology

Our analysis follows the targeted learning road map. Targeted learning is a methodology that successfully reconciles state of the art machine learning algorithms with semi-parametric inferential statistics (van der Laan & Rose, 2011). We speak of a reconciliation because, since the mid 20th century, the fields of inferential statistics and data analysis, and later machine learning, largely diverged.

It all started in the late 19th century, when probability theory began pervading nearly all scientific disciplines, a probabilistic revolution put in motion by what philosopher Hacking (1980) calls the 'erosion of determinism'. Then, in the first half-dozen decades of the 20th century, the *un*unified field of inferential statistics thrived and garnered incredible successes despite its hybrid character.

In the sixties, in reaction to the displeasing incoherence of the field, data analysis arose and gained considerable attention. Data analysis was all about the understanding and visualization of data, hunting patterns, associations and structures. In hindsight, and in light of the previous paragraph, data analysis was the consequence of the 'erosion of models', the view that all models are wrong, that the classical notion of probabilistic truth is obsolete, and that pragmatic criteria as predictive success must prevail. Nowadays, in the era of *big data* where many data-intensive empirical sciences are highly dependent on machine learning algorithms and inferential statistics, this unfortunate estrangement cannot persist.

Big data challenge both inferential statistics and machine learning. On the one hand, big data analysis requires highly scalable and efficient data manipulation (management, storage, retrieval), and requires computational efficiency of machine learning algorithms. On the other hand, it also raises new problems, pitfalls, and dif-

ficulties for statistical inference and its underlying mathematical theory. Examples include limiting the use of wrongly specified models, or acknowledging that they do not include the truth and compensating for it, if possible; the problems of small, high-dimensional data sets; the search for causal relationships in non-experimental data; the honest quantification of uncertainty; the update of efficiency theory.

Current practice all too often defines a parameter as an *un*interpretable coefficient in a misspecified parametric model (e.g., a logistic regression model) or in a small unrealistic semi-parametric regression models (e.g., a Cox proportional hazards regression), where different choices of such misspecified models yield different answers (Chambaz, Drouet, & Thalabard, 2014; van der Laan, 2015; van der Laan & Rose, 2011). In contrast, the targeted learning methodology aims at constructing confidence regions (regions which contain the truth with a certain probability, typically $95\,\%$) for user-specified *target parameters* by targeting the estimates retrieved from data-adaptive estimators (i.e., machine learning), whilst trying to rely solely on realistic statistical assumptions. This approach can reduce the divergence in outcomes of statistical analyses as model choices are automated, enabling consistent estimates regardless of the statistician performing the research.

The general road map for causal inference based on targeted learning consists of seven steps: (i) specifying knowledge about the system (i.e., what we do know about our data generating system), (ii) specifying the data, and its link to the *causal model*, (iii) specifying the target causal quantity (i.e., define what it is *exactly* that we would like to know), (iv) assessing identifiability (i.e., state and, possibly, check mild assumptions under which the target causal quantity can be inferred), (v) stating the estimation problem (i.e., defining the statistical model and what we would like to estimate, viz., estimand), (vi) performing the estimation, and (vii) interpreting the results (Petersen & van der Laan, 2014). We essentially adhere to this agenda.

8.2 HowNutsAreTheDutch

HowNutsAreTheDutch (HND) is a Dutch study that focuses on the prevalence of psychopathology, mental-health, and well-being in the Netherlands (see Chapters 3 to 5 for more details). Two of the aims of HND are to investigate the baseline heterogeneity among people in terms of psychopathology, and to raise awareness about the fact that every individual is unique (Blaauw, van der Krieke, Bos, et al., 2014; van der Krieke, Jeronimus, et al., 2016). The general philosophy behind HND is to show that psychopathology is not a binary phenomenon, and that simply classifying people (i.e., assessing whether or not someone suffers from a mental illness) based on a set of rules is difficult and error prone. In the past decades, diagnosis

and assigning classifications (or labels) of mental disorders to people has hardly had any effect on the improvement of research in psychiatry (Dehue, 2014; T. Insel, 2013; Kapur et al., 2012; Whooley, 2014). Instead, HND focuses on the graduality of psychopathology, and on the hypothesis that the combinations of various 'disorders' form the basis of the observed phenotypical expression. Furthermore, instead of merely focusing on the negative aspects of psychopathology, HND also aims to research the positive aspects of mental health, and the influence of these positive aspects on the negative ones.

HND consists of two sub-studies: (i) a cross-sectional self-report study and (ii) an intensive self-report longitudinal study. The first sub-study focuses on exploring the differences between people, and determining the distribution of various psychological traits in the Netherlands. The second sub-study focuses on the individual in specific. In this part each person measures themselves for a longer period of time, with the goal to draw conclusions on the level of the individual. In the remainder of the present work we will focus on this second sub-study.

8.2.1 The Study Protocol and Data set

In order to conduct research on the individual level, a considerable volume of data has to be available for a single individual. A well-established method of collecting such data in psychopathology research is the ecological momentary assessment (EMA) method (Shiffman, S., & Stone, 1998). EMA allows researchers to collect relatively high resolution self-report data about an individual or a group of individuals. In EMA, an individual is asked to fill out the same questionnaire for a longer period of time, which results in a time series data set of self-report questionnaire data. In the HND study the participants were asked to fill out a 43-question questionnaire three times a day for thirty successive days. Thus, each *observation* is a by product of $N = 90$ questionnaires (van der Krieke, Blaauw, et al., 2016). For the full list of questions see Table A.2 on page 202.

The HND data set used for the current analysis comprises $J = 236$ observations. For each individual, we construct *blocks* from these questionnaires, where each block contains (a subset) of the questions evaluated in each questionnaire, combined with several baseline covariates (such as age and gender). Thus, our data set is of dimension $J \times N$ (number of participants $\times$ number of blocks per participant).

We explore which activities attribute to changes in well-being. To evaluate well-being we use the construct of *positive affect* as a proxy. We measure positive affect by means of the positive affect scale from the Positive And Negative Affect Schedule (PANAS; Watson, Clark, & Tellegen, 1988). Our used implementation of the PANAS measures positive affect on ten questions on a continuous scale with a

possible range of 0 to 100. Activity is evaluated using a categorical scale that reflects which activity dominated the previous measurement period. The separate items for these measures are listed in Appendix D.1.

8.2.2 Formalization

We adhere to a well-established general mathematical representation of the variables of interest in a probabilistic framework adopted from (van der Laan & Rose, 2011). We refer to the covariates (or possible confounders) of our outcome as W, the activity measure as A (the intervention/exposure variable), and the positive affect measure as Y (the outcome). Each of these random variables is measured at a specific time t ranging over $\{1, \ldots, N\}$.

Consider an arbitrarily chosen time $t \in \{1, \ldots, N\}$. The vector $W(t)$ comprises several ambulatory variables (variables retrieved during the EMA study). These random variables can be binary, discrete, or continuous. An overview of each covariate and its type is provided in Table 8.1. We use the lowercase representation of a random variable (e.g., $w(t)$ for $W(t)$) as either a specific instance of that random variable or as an index meant to emphasize that the indexed object is intrinsically related to the corresponding random variable.

Covariate name	**Data type**	**Values**
'I am in the here and now'	Continuous	$[0, \ldots, 100]$
'How busy am I?'	Categorical	[Much too busy, Pleasantly busy, Neutral, Pleasantly quiet, Much too quiet]
'Did something special happen since the last measurement?'	Categorical	[No, Yes (positive), Yes (negative), Yes (neutral)]
'Since the last measurement moment I had a laugh'	Continuous	$[0, \ldots, 100]$
'Since the last measurement moment I was able to make a difference'	Continuous	$[0, \ldots, 100]$
'Since the last measurement moment I have been outdoors'	Continuous	$[0, \ldots, 100]$
'Since the last measurement moment I was physically active'	Continuous	$[0, \ldots, 100]$
'I did my jobs or tasks without being aware of what I was doing'	Continuous	$[0, \ldots, 100]$

Table 8.1: List of covariates used in the study.

The random variable $A(t) \in \{0, \ldots, 12\} = \mathcal{A}$ is categorical and can take one of

thirteen values. Each $a \in \mathcal{A}$ corresponds to an answer in the EMA (see Appendix D.1 for a description of the available answer categories).

Finally, the random variable $Y(t)$ is a scalar value from a continuous range of options. It ranges between 0 and 100, where 0 refers to the experience of no or a low amount of positive affect, and 100 resembles to experiencing a high amount of positive affect.

The used HND data set comprises a total of J observations. Denoted $O_1 \dots O_J$, the J observations are assumed mutually independent. This assumption is justified as access to the HND diary study was available for everyone in the Netherlands, and as such participants were included in the study at random. If a person participated in the study multiple times we only use one of their studies. Each observation O_j consists of $N = 90$ blocks, one for each time t,

$$O_j^N = ((W_j(1), A_j(1), Y_j(1)), \dots, (W_j(N), A_j(N), Y_j(N))).$$

We use the subscript j to denote that an observation belongs to a specific individual. In other words, for any individual j we have the set of N blocks.

In EMA, observations are collected chronologically, enabling us to interpret the data as a time series. It is a generally accepted fact that psychological traits at a given time can fluctuate and strongly affect psychological traits later (Rosmalen et al., 2012; van der Krieke, Blaauw, et al., 2016). An EMA reflects this sequential dependence and intraindividual variation in the sense that a block at a certain time is most probably affected by blocks preceding it. We define $\bar{O}_j(t-1)$ as all historical blocks before time t for observation j: $\bar{O}_j(t) = (O_j(s) : s \leqslant t)$.

Apart from the temporal ordering in the time series itself, we assume that an ordering exists within each block. That is, we assume that our outcome $Y(t)$ (positive affect at this exact t) is preceded by $A(t)$ (the activity one performed most of the time between $t-1$ and t) and $W(t)$, and we assume that $A(t)$ is preceded by $W(t)$. This allows us to investigate the causal relation between the variables within an observation. Although this is a strong assumption with respect to the nature of an EMA study, we will accept it for now and address it in detail in future work. The focus of the present chapter is to shed light on the application of the targeted learning methodology to EMA data, and we do not aim to draw any conclusions from such data for now, but merely show a possible use case.

We define the parents of a variable as all measured variables and historical values thereof that might influence the current value of said variable. In other words, the parents of a variable are all the measurements that precede a given variable. For each random variable at time t (given that enough historical measurements are

available), we denote these parents as follows:

$$\begin{aligned} W_j^-(t) &= \bar{O}_j(t-1), \\ A_j^-(t) &= (\bar{O}_j(t-1), W_j(t)), \\ Y_j^-(t) &= (\bar{O}_j(t-1), W_j(t), A_j(t)). \end{aligned} \tag{8.1}$$

Our objective is to derive a reliable estimate and confidence interval of a causal quantity defined as the average 'treatment' effect for any activity $a \in \mathcal{A}$ on the amount of positive affect at the next moment in time (we use the terms treatment, intervention, and activity interchangeably to be consistent with other literature). A treatment in this case is referred to as a specific activity one could perform. A reliable estimate of this average treatment effect (ATE) could inform a person of which activities they could undertake for maximizing their well-being. As we are only interested in this specific causal quantity, we can target our analysis in such a way that it will tailor and optimize the estimation for this specific focus of interest (van der Laan & Rose, 2011).

8.3 Causal and Probabilistic Perspectives

We adopt a probabilistic stance and invoke causality, a notion that plays a vital role in understanding the interwoven behavior of nature.

8.3.1 Probabilistic Framework

We view the complex natural system as a data generating process of which the exact configuration is unknown. The means we use to study this system rests on exploring the data it generates, O^N. We believe that the observations O^N can be considered as a random variable drawn from an unknown underlying true data generating system, or probability distribution. The true probability distribution (denoted P_0^N) entails infinitely many features of potential interest. One of the features of P_0^N is of special interest to us.

In practice the true data generating distribution, the conditions it applies, and features it uses to materialize O^N are unknown, and the only way we can gain knowledge about it is by reverse engineering the data observed from it, and build estimates using the empirical probability distribution based on these observations, to which we will refer to as P_N.

8.3.2 Unrealistic Ideal Experiments

Apart from the factual world, as we observe it, we are interested in *counterfactual worlds* that did not happen and were thus not observed to answer 'what-if' questions. For example, we may want to compare a counterfactual world in which an individual would have experienced $a' \neq a$ at time t to the factual world where they received a at time t instead, as it really happened. We jointly view all the worlds, factual and counterfactual, as a data generating process of which the exact configuration is unknown. We also consider as a random variable the so called *full counterfactual data* $\mathbb{O}^N$ that combines factual and counterfactual observations from all the worlds. It is drawn from a full counterfactual probability distribution denoted as $\mathbb{P}_0^N$. Note that O^N is included in $\mathbb{O}^N$, and that knowing $\mathbb{P}_0^N$ implies knowing P_0^N.

Instead of characterizing the counterfactual probability $\mathbb{P}_0^N$ which we would have liked to sample from, let us engage in the simpler task consisting in describing the *unrealistic, ideal experiment* we would have liked to carry out. It could very well be the case that the experiment is not attainable in our world.

As an illustration, say that we wish to assess to which extent performing activity $a \in \mathcal{A}$ at a certain time s attributes to positive affect at a certain time $(s+1)$. This question of interest sprouts two of them, one at the population level and the other at the individual level. For each of them we can devise an ideal experiment.

Let us consider the population level first. An ideal experiment could go as follows. A large number of times, we would repeatedly and independently (i) sample an individual from the population at time $t = 1$, (ii) follow that individual till time s, (iii) impose performing activity a at time s (an intervention), (iv) resume following the individual and report back on all covariates and the outcome at $Y(s+1)$. By taking the average outcome across all iterations we could then approximate reliably the causal effect of activity a at time s on $Y(s+1)$.

Now let us turn to the individual level. We emphasize that the ideal experiment is defined conditionally on a specific real life subject on whom we focus from time $t = 1$. The ideal experiment would go as follows. A large number of times, we would (i) follow the individual till time s, (ii) impose performing activity a at time s (an intervention), (iii) resume following the individual and report back on all the covariates and outcome at $Y(s+1)$, (iv) go back in time to time $t = 1$ and repeat, *ceteris paribus sic stantibus* (i.e., all other things being equal). Again, the average outcome across all iterations would reliably approximate the causal effect of activity a at time s on $Y(s+1)$. Reproducing this procedure for each of the activities $a \in \mathcal{A}$ would then yield approximations of the causal effects of all activities $a \in \mathcal{A}$ at time s on $Y(s+1)$. Although these hypothetical experiments are clearly impossible to perform, they do provide the conceptual basis for the present analysis.

The question remains how we can draw advantage from these unrealistic ideal experiments. The key is to relate them to our actual experiment. We do so in the so called *counterfactual framework* (Gruber & van der Laan, 2009; Pearl, 2009; Rubin, 1974), viewing each observed, real data as a fraction of the full counterfactual data. The complete data consists of all the above counterfactual $Y(s+1)$ obtained by imposing activity a at time s, *ceteris paribus sic stantibus*. The real data reduces to the parts of the complete data which corresponds to the activity $A(t)$ that was observed at time t. We thus frame our analysis in terms of missingness.

8.3.3 Causal Model, Counterfactuals, and Quantity

We define our causal model using the earlier defined notation and knowledge we have about the underlying data-generating process. By defining a causal model that captures this knowledge, we formalize and express our assumptions about this data-generating process (Duncan, 1975; Pearl, 2009; Petersen & van der Laan, 2014).

We frame our presentation in terms of a nonparametric structural equation model (NPSEM). Let us first introduce a NPSEM for the generation of our real observations. We assume the existence of $3N$ deterministic functions $f_{w(t)}$, $f_{a(t)}$, and $f_{y(t)}$ ($t = 1, \ldots, N$) and, for every $1 \leqslant j \leqslant J$, of a source of randomness

$$(U_{j,w(1)}, U_{j,a(1)}, U_{j,y(1)}, \ldots, U_{j,w(N)}, U_{j,a(N)}, U_{j,y(N)})$$

such that, for each $1 \leqslant j \leqslant J$, the random generation of O_j^N decomposes as follows: for $t = 1, \ldots, N$, sequentially,

$$\begin{aligned} W_j(t) &= f_{w(t)}(W_j^-(t), U_{j,w(t)}), \\ A_j(t) &= f_{a(t)}(A_j^-(t), U_{j,a(t)}), \\ Y_j(t) &= f_{y(t)}(Y_j^-(t), U_{j,y(t)}). \end{aligned} \tag{8.2}$$

It is important to note that in this abstract representation we give no explicit form to the deterministic mechanisms $f_{w(t)}$, $f_{a(t)}$, and $f_{y(t)}$ (hence the NP in NPSEM). The components of the source of randomness are exogenous variables which drive the generation of $W_j(t)$, $A_j(t)$, and $Y_j(t)$, but were not measured by the current EMA study. Note that that the mechanisms $f_{w(t)}$, $f_{a(t)}$, and $f_{y(t)}$ ($t = 1, \ldots, N$) in Equation (8.2) are shared among all J observed individuals. The assumption that all individuals are subject to the same data generating process is justified by the belief that nature behaves in this way. It allows us to combine all observations to estimate certain features of the common data generating distribution, instead of solely basing them on data specific to the individual.

We now use the model from Equation (8.2), which produces $O_j^N \sim P_0^N$, to define our causal model, which produces $\mathbb{O}_j^N \sim \mathbb{P}_0^N$. For every $1 \leqslant s < N$ and $a \in \mathcal{A}$, for $t = 1, \ldots, N$, sequentially, let

$$\begin{aligned} W_{j,s,a}(t) &= f_{w(t)}(W^-_{j,s,a}(t), U_{j,w(t)}), \\ A_{j,s,a}(t) &= \begin{cases} a \text{ if } s = t, \\ f_{a(t)}(A^-_{j,s,a}(t), U_{j,a(t)}) \text{ if } t \neq s, \end{cases} \\ Y_{j,s,a}(t) &= f_{y(t)}(Y^-_{j,s,a}(t), U_{j,y(t)}), \end{aligned} \tag{8.3}$$

then let the (j, s, a)-specific complete data $\mathbb{O}^N_{j,s,a}$ be obtained by combining the outputs of Equation (8.3),

$$\mathbb{O}^N_{j,s,a} = ((W_{j,s,a}(1), A_{j,s,a}(1), Y_{j,s,a}(1)), \ldots, (W_{j,s,a}(N), A_{j,s,a}(N), Y_{j,s,a}(N))).$$

The j-specific complete data $\mathbb{O}_j^N$ is obtained by combining the (j, s, a)-specific complete data $\mathbb{O}^N_{j,s,a}$ across $1 \leqslant s < N$ and $a \in \mathcal{A}$. In Equation (8.3), each $W^-_{j,s,a}(t)$, $A^-_{j,s,a}(t)$ and $Y^-_{j,s,a}(t)$ are extracted from $\mathbb{O}^N_{j,s,a}$ just like $W_j^-(t)$, $A_j^-(t)$ and $Y_j^-(t)$ are extracted from O_j, see Equation (8.1). In the two above displays, we use two subscripts to identify the counterfactual data distribution: we use s to denote that the distribution received an intervention at time s and we use a to denote that during this intervention activity $A(s) = a$ was imposed. Using this NPSEM we can impose any activity $a \in \mathcal{A}$ at any time $s = 1, \ldots, N-1$ and collect the outcome of $Y(s+1)$. As previously noted, sampling independently and a large number of times from Equation (8.3) and averaging average across all the resulting outcomes at time $(s + 1)$ would approximate our quantity of interest.

We are now in a position to give a formal definition of our causal quantities. For every $1 \leqslant j \leqslant J$, each $s = 1, \ldots, N - 1$ and every $a \in \mathcal{A}$, the corresponding causal quantity of interest is

$$\text{CQ}^N_{0,j,s,a} = \mathbb{E}_{\mathbb{P}_0^N}(Y_{j,s,a}(s + 1) \mid W_{j,s,a}(1)). \tag{8.4}$$

It quantifies the effect for individual j of imposing activity a at time s as measured in terms of the average outcome at time $(s + 1)$. Note that $\text{CQ}^N_{0,j,s,a}$ is still random because it is defined conditionally on $W_{j,s,a}(1)$. However, the expectation $\mathbb{E}_{\mathbb{P}_0^N}[\text{CQ}^N_{0,j,s,a}]$ (where we integrate out $W_{j,s,a}(1)$ with respect to its distribution in the population) does not depend on j, is therefore deterministic, and should be interpreted as the average causal effect in the population of imposing activity a at time s, measured at time $(s + 1)$.

8.4 Statistical Model

We define the statistical model as the collection $\mathcal{M}$ of all possible probability distributions that nature could have used to generate O^N. Note that $O^N \sim P_0^N$, and $P_0^N \in \mathcal{M}$, meaning that by definition the statistical model contains the true probability distribution. In contrast, if we were to assume that $P_0^N \in \mathcal{M}^\theta \subset \mathcal{M}$, where $\mathcal{M}^\theta$ is a parametric model, and in reality this is not the case, we would obtain a misspecified statistical model with unreliable parameter estimates. A realistic statistical model reflects only true knowledge and we should not impose any unfounded restrictions on it.

8.4.1 Nonparametric Statistical Model

In a nonparametric statistical model no parametric assumptions are made on the statistical model. That is, we do not assume that the statistical model can be represented by a finite (low) dimensional number of parameters. By assuming the statistical model to be nonparametric, we circumvent making any assumptions that do not represent knowledge, and as such yielding a statistical model that is too restrictive or misspecified. Essentially, a nonparametric model can be represented using an infinite number of parameters. In order to work with such a model, we do need to introduce several assumptions, as laid out in the next section.

Main assumptions.

Fix arbitrarily $1 \leqslant j \leqslant J$. For every $t = 1, \ldots, N$, the random variables $W_j(t)$, $A_j(t)$, and $Y_j(t)$ each depend on their respective parents $W_j^-(t)$, $A_j^-(t)$, and $Y_j^-(t)$, as shown in Equation (8.1), a sequential dependence that naturally occurs in a time series. Although the parents of a variable entail all past values, we cannot handle such complexity. In fact, the number of parameters in our model would have to grow at the same rate as t, and would cause our estimation problem to become intractable as theoretically each observation could have its own distribution (Benkeser et al., 2016). We therefore introduce three assumptions with respect to the relevant past of these random variables and their data generating process: (i) a stationarity assumption, (ii) a Markov-type assumption, and (iii) a nesting assumption on the summary measures.

With the stationarity assumption, we postulate that there exist shared mechanisms over time t. From a practical point of view, it allows us to combine observations over time to base our estimators on (van der Laan & Rose, 2017), and eventually enables making predictions and estimating.

The Markov-type and nesting assumptions state the existence of $3N$ deterministic functions $\gamma_{w,t}$, $\gamma_{a,t}$, $\gamma_{y,t}$ (not indexed by j) taking values in $\mathbb{R}^{d_w}$, $\mathbb{R}^{d_a}$, $\mathbb{R}^{d_y}$ ($t = 1, \ldots, N$) such that, for every $t = 1, \ldots, N$, $W_j(t)$, $A_j(t)$, and $Y_j(t)$ depend on their full histories $W_j^-(t)$, $A_j^-(t)$, $Y_j^-(t)$ only through the fixed-dimensional summary measures

$$\begin{aligned} W_{j,c}^-(t) &= \gamma_{w,t}(W_j^-(t)), \\ A_{j,c}^-(t) &= (\gamma_{a,t}(A_j^-(t)), W_j(t), W_{j,c}^-(t)), \\ Y_{j,c}^-(t) &= (\gamma_{y,t}(Y_j^-(t)), A_j(t), A_{j,c}^-(t)), \end{aligned} \tag{8.5}$$

respectively. In view of Equation (8.2), this assumption essentially boils down to postulating that the $3N$ deterministic functions $f_{w(t)}$, $f_{a(t)}$, and $f_{y(t)}$ ($t = 1, \ldots, N$) can be replaced by three deterministic functions of the form $f \circ \gamma$ where γ extracts the relevant information from the history and f processes it.

Examples of such summary measures include the p most recent observations (lags) of data; the running mean of the historical data; or any measure that is relevant and could be represented using a metric that does not depend on data with a dimensionality growing with t. Importantly, it may be necessary to shift to the right the initial time (currently, $t = 1$) to use consistently some summary measures. For instance, if $\gamma_{w,t}$, $\gamma_{a,t}$ and $\gamma_{y,t}$ were chosen so that $W_{j,c}^-(t) = (O_j(t-1), \ldots, O_j(t-p))$, $A_{j,c}^-(t) = (W_j(t), O_j(t-1), \ldots, O_j(t-p))$ and $Y_{j,c}^-(t) = (A_j(t), W_j(t), O_j(t-1), \ldots, O_j(t-p))$ for some fixed (i.e., independent of N) $1 \leqslant p \leqslant N$, then the initial time should be set at $t = (p+1)$ and the final time at $t = N - p$. Formally, this can be done elegantly by allowing the generic random variable $W(1)$ to differ in structure from the generic random variables $W(2), \ldots, W(N)$ so as to include in $W(1)$ what we would then denote $(O(-p+1), \ldots, O(0))$.

True likelihood.

Under the above assumptions, knowing P_0^N is the same as knowing three conditional densities. Specifically, introducing

$$\begin{aligned} W_c^-(t) &= \gamma_{w,t}(W^-(t)), \\ A_c^-(t) &= (\gamma_{a,t}(A^-(t)), W(t), W_c^-(t)), \\ Y_c^-(t) &= (\gamma_{y,t}(Y^-(t)), A(t), A_c^-(t)) \end{aligned} \tag{8.6}$$

(for $t = 1, \ldots, N$), knowing P_0^N is the same as knowing the conditional densities $\bar{q}_{0,w}$, $\bar{g}_0$ and $\bar{q}_{0,y}$ of $W(t)$ given $W_c^-(t)$, $A(t)$ given $A_c^-(t)$ and $Y(t)$ given $Y_c^-(t)$, respectively (any $t \in \{1, \ldots, N\}$). Thus, the (true) likelihood of O_j^N under P_0^N can be

expressed as

$$\begin{aligned} P_0^N(O_j^N) = &\prod_{t=1}^{N} \bar{q}_{0,w}(W_j(t) \mid W_{j,c}^-(t)) \\ &\times \prod_{t=1}^{N} \bar{g}_0(A_j(t) \mid A_{j,c}^-(t)) \\ &\times \prod_{t=1}^{N} \bar{q}_{0,y}(Y_j(t) \mid Y_{j,c}^-(t)). \end{aligned} \tag{8.7}$$

The conditional densities $\bar{q}_{0,w}$, $\bar{g}_0$ and $\bar{q}_{0,y}$ are considered to be elements of nonparametric spaces $\mathcal{Q}_w$, $\mathcal{G}$, $\mathcal{Q}_y$.

8.4.2 Counterfactual Nonparametric Statistical Model

We can construct the counterfactual likelihood by combining the likelihood as defined in Equation (8.7) with the counterfactual mechanisms defined in Equation (8.3). We consider the case that we impose activity $a \in \mathcal{A}$ at time $1 \leqslant s < N$.

Let $\mathbb{I}\{\cdot\}$ be the indicator function returning 1 if the expression inside the brackets holds and 0 otherwise. Let $\mathbb{P}_{0,s,a}^N$ be the marginal joint distribution of $\mathbb{O}_{j,s,a}$ as it is derived from the distribution $\mathbb{P}_0^N$ of the complete data $\mathbb{O}_j^N$. The counterfactual likelihood of O_j^N under $\mathbb{P}_{0,s,a}$ equals

$$\begin{aligned} \mathbb{P}_{0,s,a}^N(O_j^N) = &\prod_{t=1}^{N} \bar{q}_{0,w}(W_j(t) \mid W_{j,c}^-(t)) \\ &\times \prod_{\substack{t=1 \\ t \neq s}}^{N} \underbrace{\left(\bar{g}_0(A_j(t) \mid A_{j,c}^-(t))\right)}_{\text{measured}} \times \underbrace{\mathbb{I}\{A_j(s) = a\}}_{\text{counterfactual}} \\ &\times \prod_{t=1}^{N} \bar{q}_{0,y}(Y_j(t) \mid Y_{j,c}^-(t)). \end{aligned} \tag{8.8}$$

Equation (8.8) is known as the *G-Computation formula* (Gill & Robins, 2001).

8.4.3 Target Statistical Parameter

It is now time to define our target statistical parameters, built as statistical versions of the causal quantities of interest introduced earlier in Section 8.3.3. The target parameters are said *statistical* to emphasize that, contrary to the causal quantities of interest, they can be viewed as functions of P_0^N (whereas the causal quantities of interest are functions of $\mathbb{P}_0^N$).

Consider the case that we impose activity $a \in \mathcal{A}$ at time $1 \leqslant s < N$. Fix arbitrarily $1 \leqslant j \leqslant J$. The corresponding target statistical parameter is defined as $\psi^N_{j,s,a}(P^N_0)$, where the mapping $\psi^N_{j,s,a} : \mathcal{M} \to \mathbb{R}$ is given by

$$\psi^N_{j,s,a}(P^N) = \mathbb{E}_{P^N}\left[Y_j(s+1) \times Z_{j,s,a} \mid W_j(1)\right], \tag{8.9}$$

where in turn the so called change of probability $Z_{j,s,a}$ equals

$$Z_{j,s,a} = \frac{\mathbb{I}\{A_j(s) = a\}}{\bar{g}(A_j(s) \mid A^-_{j,c}(s))} \tag{8.10}$$

($\bar{g}$ is the conditional density of $A(N)$ given $A^-_c(N)$ under P^N).

The question that remains open is how $\psi^N_{j,s,a}(P^N_0)$ relates to $\mathrm{CQ}^N_{0,j,s,a}$. To answer this question, let us now introduce three key assumptions, the so called (i) *consistency*, (ii) *positivity* and (iii) *sequential randomization* assumptions.

First, the consistency assumption states that O^N_j equals $\mathbb{O}_{j,s,A_j(s)}$ for every $1 \leqslant s < N$. It is met by construction because of the way Equation (8.3) is built around Equation (8.2).

Second, the positivity assumption states that at each time $1 \leqslant t \leqslant N$, all activities have a conditional probability uniformly bounded away from zero given the parents $A^-_j(t)$ of an activity. Formally, there exists $\delta > 0$ such that for every $1 \leqslant t \leqslant N$, for all $a \in \mathcal{A}$,

$$\bar{g}_0(a \mid A^-_j(t)) = P^N_0(A_j(t) = a \mid A^-_j(t)) \geqslant \delta.$$

In words, it must not be the case that at a certain point of an individual trajectory (i.e., during the progressive construction of an O^N_j), one or more activities surely cannot be performed.

Third, the sequential randomization assumption states that for all time $1 \leqslant t \leqslant N$, the activity $A_j(t)$ at that time is conditionally independent from the potential outcomes $\{Y_{j,s,a}(t) : a \in \mathcal{A}\}$ given the parents $A^-_j(t)$ of activity. In view of Equation (8.2) and Equation (8.3), this concerns the distribution of source of randomness. In words, the assumption excludes the existence of unmeasured confounders.

Under these assumptions, it can be shown that the target statistical parameter identifies its causal counterpart. Formally, it holds that

$$\psi^N_{j,s,a}(P^N_0) \equiv \mathrm{CQ}^N_{0,j,s,a}. \tag{8.11}$$

8.5 Online Targeted Learning

So far, EMA data have been investigated using methods based on finite-dimensional parametric models, including multilevel regression (e.g., Jahng, Wood, & Trull,

2008; H. M. Schenk, Bos, Slaets, de Jonge, & Rosmalen, 2017), or vector autoregression (e.g., Emerencia et al., 2016; Rosmalen et al., 2012; Snippe et al., 2015; van Gils et al., 2014). The latter is generally used to derive causality in the form of *Granger causality* (Granger, 1969).

The parametric (and typically linear) nature of these statistical models makes the corresponding $\mathcal{M}^\theta$s very small sub models of the nonparametric model $\mathcal{M}$ where our targeted analysis takes place. Therefore, blindly assuming that the parametric statistical models are well specified causes issues in their interpretation when they are in fact misspecified, which is certainly almost always the case (e.g., Neugebauer et al., 2013).

Once a model (parametric or not) has been chosen, the procedure consisting of learning P_0^N based on data sampled from it and using the model can be viewed as the evaluation of an *algorithm* at the empirical distribution. This algorithmic interpretation makes it easy to contrast a statistical method based on a finite-dimensional model and the targeted learning method that we develop in the present work. Whereas the former uses a fixed algorithm, the latter data-adaptively builds a meta-algorithm out of a collection of prescribed algorithms based on the cross-validation (CV) principle. Taking typically the form of a convex combination of algorithms, the meta-algorithm data-adaptively puts more weight on these algorithms that perform well and less on the others.

8.5.1 Overview

Resorting to more flexible, data-adaptive algorithms to learn P_0^N is only one of the two features that contrast targeted learning from learning based on a finite-dimensional model. Once P_0^N has been learned (i.e., once we have built estimators of $\bar{q}_{0,w}$, $\bar{g}_0$, and $\bar{q}_{0,y}$), we can derive an estimator of $\psi_{j,s,a}^N(P_0^N)$ by mere substitution. However, such an estimator does not lend itself well to the construction of a confidence interval because, upstream, the meta-algorithms that produced the estimator of P_0^N relied on optimality criteria indifferent to the fact that it would eventually be used to estimate $\psi_{j,s,a}^N(P_0^N)$. It remains to target them.

The distinction we just made is encapsulated in the motto *"predicting is not explaining"* (Chambaz & Desagulier, 2015; Shmueli, 2010). The concept of targeting is nicely explained using a simple example. Suppose you are a chef in a restaurant and you have an apprentice. You tell the apprentice that you will ask him or her to cook a very complex dish the next day, but you do not say what the dish is. The apprentice will probably study all night on preparing various dishes and cooking techniques in order to be able to be well prepared for any dish. The next day the apprentice makes the dish of your choice and the result is reasonable. Now suppose

you would have instead asked the apprentice to learn cooking a specific dish (the 'target' dish). This would drastically change the way s/he would prepare, as the apprentice his or her learning would probably be targeted towards the target dish, which is also reflected in the final result.

In conclusion, the targeted learning methodology that we develop here unfolds in two steps. The first step consists in estimating P_0^N using flexible, data-adaptive algorithms. The second step consists in estimating the estimator of P_0^N to acknowledge the fact that it is eventually exploited to estimate $\psi_{j,s,a}^N(P_0^N)$ and build a confidence interval around its estimator.

8.5.2 On Machine Learning of Infinite-dimensional Features

We understand machine learning as the process consisting in training an algorithm on data for the sake of making reliable predictions on new, unseen data. With this statement, it is apparent that the statistical task of predicting is included in machine learning. In broad terms, machine learning goes beyond in the sense that it adopts more aggressive, data-adaptive strategies at the cost of paying less (if not neglecting) attention to the construction of confidence bands around the predictions.

Casting the presentation of machine learning in a simple framework.

In order to make predictions, one typically estimates a regression function or the data generating distribution itself. For clarity, in this section we focus on a small example model.

Let $Z_1 = (X_1, Y_1), \ldots, Z_n = (X_n, Y_n), Z_{n+1} = (X_{n+1}, Y_{n+1})$ be drawn independently from the unknown probability distribution Π_0 and say that our objective is to learn to predict the outcome Y_{n+1} based on $Z_1, \ldots, Z_n$ and X_{n+1}. Let $\mathcal{Z} = \mathcal{X} \times]0,1[$ be the space where a generic $Z = (X, Y)$ sampled from Π_0 takes its values (what follows would be unaltered if we replaced $]0,1[$ with $\{0,1\}$). Let us assume that $\mathcal{Z}$ is known to us. In the absence of further knowledge on Π_0, introduce the nonparametric space $\mathcal{P}$ of all probability distributions on $\mathcal{Z}$. In particular, $\Pi_0 \in \mathcal{P}$.

Say that we engage in the machine learning of the conditional expectation f_0 of Y given X under Π_0. An infinite-dimensional feature of Π_0, f_0 is characterized by $f_0(X) = \mathbb{E}_{\Pi_0}(Y \mid X)$. It belongs to the set $\mathcal{F}$ of all functions mapping $\mathcal{X}$ to $]0,1[$.

Let $\mathcal{P}^e \subset \mathcal{P}$ be the subset of $\mathcal{P}$ defined as

$$\mathcal{P}^e = \bigcup_{M \geqslant 1} \left\{ \frac{1}{M} \sum_{m=1}^{M} \mathrm{Dirac}_{z_m} : z_1, \ldots, z_M \in \mathcal{Z} \right\},$$

where Dirac_z is the probability distribution that puts all its mass on $\{z\}$. We think of $\mathcal{P}^e$ as the subset of all possible empirical distributions in $\mathcal{P}$ (hence the superscript

e). In particular, the empirical distribution

$$P_n = \frac{1}{n}\sum_{i=1}^{n} \text{Dirac}_{Z_i}$$

that we do observe satisfies $P_n \in \mathcal{P}^e$.

We formalize the machine learning of f_0 as any mapping Φ from $\mathcal{P}^e$ to $\mathcal{F}$ associated with a valid loss function L for f_0. A valid loss function L for f_0 is a function mapping $\mathcal{F}$ to $\mathbb{R}^{\mathcal{Z}}$ (the set of real-valued functions over $\mathcal{Z}$) in such a way that

$$f_0 = \underset{f\in\mathcal{F}}{\arg\min}\, \mathbb{E}_{\Pi_0}\left(L(f)(Z)\right). \tag{8.12}$$

For instance, L could be the least-square loss function L_2 given by

$$L_2(f)(Z) = (Y - f(X))^2,$$

or the logistic loss function L_1 given by

$$L_1(f)(Z) = -Y \log f(X) - (1-Y)\log(1 - f(X)).$$

In both cases, $L(f)(Z)$ is a metric of the quality of the predictions of Y by $f(X)$ that contrasts the prediction with the actual outcome. As for Φ, it could be characterized by

$$\Phi(p) = \underset{f\in\mathcal{F}(p)}{\arg\min}\, \mathbb{E}_{p}\left(L(f)(Z)\right) \tag{8.13}$$

for every $p \in \mathcal{P}^e$, where $\mathcal{F}(p) \subset \mathcal{F}$ is a p-dependent subset of $\mathcal{F}$ (typically, its 'size' may depend on the number of observations upon which p is based). Note how we substitute p and $\mathcal{F}(p)$ for Π_0 and $\mathcal{F}$ in Equation (8.13) relative to Equation (8.12).

It is known that we cannot assess the performance of $\Phi(P_n)$ as an estimator of f_0 based on its empirical risk

$$\mathbb{E}_{P_n}\left(L(\Phi(P_n))(Z)\right) = \frac{1}{n}\sum_{i=1}^{n} L(\Phi(P_n))(Z_i),$$

because that would necessarily be overoptimistic (Chambaz & Desagulier, 2015). In fact, it is generally bad practice to use the same data for both training and testing our estimator. To see this, it suffices to consider an algorithm Φ trained to output Y_i at each X_i and to sample Y from the uniform distribution on $]0,1[$ at any $X \notin \{X_1,\ldots,X_n\}$. Its performance would be perfect on the training set, but disastrous on new data. We also refer to the work of Bottou, Curtis, and Nocedal (2016) for another example. We thus need a technique to assess how well an algorithm Φ performs. A crucial issue cross-validation solves.

Cross-validation.

Cross-validation (CV) is a sample-splitting technique consisting in splitting up the empirical data $\{Z_1, \ldots, Z_n\}$ into a training and a validation data set based on a splitting random variable B_n. In V-fold CV, B_n can take V different values $b_1, \ldots, b_V \in \{0,1\}^n$. Each realization b_v corresponds to a splitting into the training data set $\{Z_i : 1 \leqslant i \leqslant n, b_v(i) = 0\}$ and the validation data set $\{Z_i : 1 \leqslant i \leqslant n, b_v(i) = 1\}$ (here, $b_v(i)$ is the i^{th} entry of vector b_v). By choice, the entries of the vector $\sum_{v=1}^{V} b_v$ are all equal to one. In other words, every observation falls only once in a validation set. Moreover, the proportion of $n^{-1}\sum_{i=1}^{n} b_v(i)$ of entries of b_v that equal one is approximately V^{-1} for all $1 \leqslant v \leqslant V$.

Denote P^0_{n,B_n} and P^1_{n,B_n} the empirical distributions of the training and validation data sets, respectively. The algorithm is trained on P^0_{n,B_n}, and its predictive qualities are evaluated on P^1_{n,B_n}. The *cross-validated risk* of Φ is defined as

$$R_{L,B_n}(\Phi) = \mathbb{E}_{B_n}\left[\mathbb{E}_{P^1_{n,B_n}}\left(L(\Phi(P^0_{n,B_n})(Z))\right)\right]. \tag{8.14}$$

It is a sensible assessment of how well algorithm Φ performs the machine learning of f_0. Note how the algorithm is trained on P^0_{n,B_n} on the one hand, and evaluated on P^1_{n,B_n} on the other hand.

Ensemble learning.

There exists a plethora of learning algorithms and, currently, no single learning algorithm outperforms all the others across all learning tasks (Wolpert, 1996). Moreover, it is impossible to determine *a priori* which algorithm from a collection of algorithms will perform best in a given context. Thus, it is largely admitted that it is better to capitalize on the collection instead of choosing beforehand one single algorithm from it (e.g., Dietterich, 2000; Gashler, Giraud-Carrier, & Martinez, 2008; Lemke, Budka, & Gabrys, 2015).

Capitalizing means either to data-adaptively identify and select the best algorithm from the collection for the problem at hand, or to data-adaptively build the best combination of algorithms for the problem at hand (Chambaz & Desagulier, 2015). Combining multiple learners into a *metalearner* is generally known as *ensemble learning* (Lemke et al., 2015).

One specific ensemble learning algorithm is the *SuperLearner* algorithm (Polley, Rose, & van der Laan, 2011; van der Laan, Polley, & Hubbard, 2007), an instance of a so called 'stacking' ensemble method. Let $\Phi_1, \ldots, \Phi_K$ be K individual machine learning algorithms associated with the same valid loss function L for f_0. For each $1 \leqslant k \leqslant K$, the performance of algorithm Φ_k is assessed through its cross-validated

risk $R_{L,B_n}(\Phi_k)$. Defining

$$k_n = \underset{1 \leqslant k \leqslant K}{\arg\min}\, R_{L,B_n}(\Phi_k),$$

SuperLearning identifies Φ_{k_n} as the metalearner to employ. The resulting estimator of f_0 is thus $\Phi_{k_n}(P_n)$.

How well algorithm Φ_{k_n} works is established through a so called *oracle inequality*. Generally speaking, an oracle inequality compares the performance of any $\Phi \in \{\Phi_1, \ldots, \Phi_K\}$ (Φ possibly random, like Φ_{k_n}) to the performance of the oracle $\Phi_{k_{0,n}}$, with

$$k_{0,n} = \underset{1 \leqslant k \leqslant K}{\arg\min}\, R_{0,L,B_n}(\Phi_k)$$

where the oracle cross-validated risk is given by

$$R_{0,L,B_n}(\Phi_k) = \mathbb{E}_{B_n}\left[\mathbb{E}_{\Pi_0}\left(L(\Phi(P^0_{n,B_n}))(Z)\right)\right]. \tag{8.15}$$

Comparing Equation (8.14) and Equation (8.15) reveals that the sole difference between $R_{L,B_n}(\Phi_k)$ and $R_{0,L,B_n}(\Phi_k)$ is that the inner expectation is relative to P^1_{n,B_n} in the former and to Π_0 in the latter. Heuristically, the oracle cross-validated risk $R_{0,L,B_n}(\Phi_k)$ is thus a more reliable measure of performance of Φ_k than its counterpart $R_{L,B_n}(\Phi_k)$ because it benefits from the use of the true Π_0 as opposed to its empirical counterpart P^1_{n,B_n}.

Going back to the oracle inequality for the metalearner Φ_{k_n}, it takes the following form: for any $\delta \in]0,1[$, there exists $c_1(\delta) > 0$ such that, with probability $(1-\delta)$,

$$\begin{aligned} 0 &\leqslant R_{0,L,B_n}(\Phi_{k_n}) - R_{0,L,B_n}(\Phi_{k_{0,n}}) \\ &\leqslant (1 + c_1(\delta)) \min_{1 \leqslant k \leqslant K} \left\{R_{0,L,B_n}(\Phi_k) - R_{0,L,B_n}(\Phi_{k_{0,n}})\right\} + \text{Remainder}. \end{aligned} \tag{8.16}$$

Typically, $\delta \mapsto c_1(\delta)$ decreases: the smaller is δ, the more likely the inequality is verified, but the looser the comparison is, because $c_1(\delta)$ is larger. Sometimes, however, it is possible to guarantee that $c_1(\delta) = 0$, in which case the inequality is considered tight. As for the remainder term, it depends on δ, K and n. Of course, the smaller is the remainder term, the more meaningful is the oracle inequality. In many contexts (Benkeser et al., 2016; van der Laan et al., 2007), the remainder term takes the form

$$c_2(\delta)\frac{\log(K)}{n^\alpha},$$

for some constant $c_2(\delta) > 0$ and coefficient $\alpha \in]0,1]$ that depends on loss function L (Polley et al., 2011; van der Laan & Dudoit, 2003). The remarkable feature of

this expression is that the number K of algorithms composing the collection plays a role through its logarithm $\log(K)$. Therefore, it is possible to let K increase with n as long as $\log(K)/n^{\alpha}$ still goes to zero sufficiently fast. For instance, a polynomial dependence is allowed.

We just described the data-adaptive identification (by k_n) and selection of the best single algorithm in the collection (i.e., Φ_{k_n}). The last remark prompts us to go beyond this *single* algorithm and look for the best *combination* of algorithms from the collection. Formally, let

$$\Omega_n \subset \Omega_\infty = \left\{ \omega \in \mathbb{R}_+^K : \sum_{k=1}^K \omega_k = 1 \right\}$$

be an (n^{-1})-net of cardinality $\mathcal{O}(n^K)$ such that (i) it contains the K vectors whose coordinates are all equal to 0 except for one which equals 1, and (ii) for every $\omega \in \Omega_\infty$, there exists $\omega_n \in \Omega_n$ for which $\|\omega - \omega_n\| \leqslant n^{-1}$. One should think of Ω_n as a deterministic approximation to Ω_∞, whose elements are weights (they are nonnegative and sum up to one). Now, introduce the collection of

$$\left\{ \Phi_\omega = \sum_{k=1}^K \omega_k \Phi_k : \omega \in \Omega_n \right\} \supset \{\Phi_1, \ldots, \Phi_K\}.$$

A generic element Φ_ω of the above collection is a metalearner, i.e., a mapping from $\mathcal{P}^e$ to $\mathcal{F}$ such that $\Phi_\omega(P_n)$ estimates the targeted f_0. If we define

$$\omega_n = \underset{\omega \in \Omega_n}{\arg\min}\, R_{L,B_n}(\Phi_\omega),$$

then the above discussion shows that the empirical metalearner Φ_{ω_n} performs almost as well as the oracle $\Phi_{\omega_{0,n}}$, where

$$\omega_{0,n} = \underset{\omega \in \Omega_n}{\arg\min}\, R_{0,L,B_n}(\Phi_\omega).$$

This remarkable result expresses in words what is formally described as an oracle inequality, very much similar to Equation (8.16). Note that the minimum in the new oracle inequality ranges over a much bigger set than in Equation (8.16), and that the combination of algorithms Φ_{ω_n} necessarily outperforms the single algorithm Φ_{k_n}.

Online learning.

Data sets nowadays can be so large that they require highly scalable and efficient machine learning algorithms. Moreover, data is often collected via streams, and as such, accumulates over time. In particular, it is often impossible (computationally

and data-storage-wise) to process all the data in one single step, as *batch learning* algorithms do (Hastie et al., 2009). Instead, novel *out-of-core* (or online) algorithms need to be applied. As opposed to batch learning, *online learning* is a method that promises a way to cope with such big data. In online learning, an estimator is trained in iterations on small subsets ('mini-batches') of the full data set, possibly as they come in (if not all data is available *a priori*; Hastie et al., 2009).

More formally, with online learning, an algorithm $\Phi(\bar{O}(t))$ can be updated with a sample of size $1 \leqslant n \ll N$ to create an updated algorithm $\Phi(\bar{O}(t+n))$. The advantage of online learning is that it allows for iteratively updating the (initial) fit of an estimator. That is, after an estimator has been trained on an initial subset of the data, online learning offers a means to iteratively update this fit with new data. As such, online learning algorithms do not need to retain all data, and can perform an update step based on a small subset. Training of the estimator can therefore be performed in an iterative fashion, in which each iteration requires a relatively small amount of memory and computing power.

8.5.3 Step one: infinite-dimensional features

The first step of our targeted learning methodology consists in the machine learning of the three conditional densities $\bar{q}_{0,w}$, $\bar{g}_0$ and $\bar{q}_{0,y}$. They are evidently infinite-dimensional features of P_0^N and, in fact, as we already remarked in Section 8.4.1, they altogether describe completely P_0^N. To learn these conditional densities, we choose the negative log-likelihood loss function ℓ. For any $\bar{q}_w \in \mathcal{Q}_w$, $\bar{g} \in \mathcal{G}$ and $\bar{q}_y \in \mathcal{Q}_y$, for any $1 \leqslant j \leqslant J$ and $1 \leqslant t \leqslant N$, we have

$$\ell(\bar{q}_w)(O_j(t)) = -\log \bar{q}_w(W_j(t) \mid W_{j,c}^-(t)), \tag{8.17}$$

$$\ell(\bar{g})(O_j(t)) = -\log \bar{g}(A_j(t) \mid A_{j,c}^-(t)), \tag{8.18}$$

$$\ell(\bar{q}_y)(O_j(t)) = -\log \bar{q}_y(Y_j(t) \mid Y_{j,c}^-(t)). \tag{8.19}$$

In view of Equation (8.7), the log-likelihood of O_j^N under $P^N \in \mathcal{M}$ characterized by the triplet $(\bar{q}_w, \bar{g}, \bar{q}_y)$ then writes as

$$\log P^N(O_j^N) = -\sum_{t=1}^{N} \left(\ell(\bar{q}_w)(O_j(t)) + \ell(\bar{g})(O_j(t)) + \ell(\bar{q}_y)(O_j(t))\right). \tag{8.20}$$

We have to adopt this particular loss function because inferring the target parameter $\psi_{j,s,a}^N(P^N)$, see Equation (8.9), requires that we estimate the full likelihood as opposed to well chosen conditional expectations.

The learning task is very challenging, not so much because of the conditional density $\bar{g}_0$, but mainly because of the conditional densities $\bar{q}_{0,w}$ and $\bar{q}_{0,y}$. Since $\mathcal{A} =$

$\{0, \ldots, 12\}$, learning $\bar{g}_0$ boils down to estimating twelve conditional probabilities. Specifically, because observing $A(t)$ is equivalent to observing

$$(\mathbb{I}\{A(t) \leqslant 0\}, \mathbb{I}\{A(t) \leqslant 1\}, \ldots, \mathbb{I}\{A(t) \leqslant 11\}),$$

knowing $\bar{g}_0$ is equivalent to knowing

$$P_0^N(A(t) \leqslant a \mid A_c^-(t), A(t) \geqslant a) \tag{8.21}$$

for all $0 \leqslant a \leqslant 11$. Therefore, building an estimator of $\bar{g}_0$ is equivalent to constructing twelve estimators of Equation (8.21) for all $0 \leqslant a \leqslant 11$. In contrast, estimating the conditional densities $\bar{q}_{0,w}$ and $\bar{q}_{0,y}$ is difficult because (i) the spaces where $W_c^-(t)$ and $Y_c^-(t)$ live (every $1 \leqslant t \leqslant N$) are large, and (ii) $W(t)$ and $Y(t)$ are not discrete. We thus decide to apply a *discretization method* (Munoz & van der Laan, 2011; Sofrygin & van der Laan, 2017). Specifically, we discretize every coordinate of $W(t)$ and $Y(t)$ itself (all $1 \leqslant t \leqslant N$) using bins. Focusing on $Y(t)$, which is simpler because $Y(t)$ is one-dimensional, we substitute for $Y(t)$ a set of indicators

$$\{\mathbb{I}\{Y(t) \in \cup_{l' \leqslant l} b_{l'}\} : 1 \leqslant l < l_y\}$$

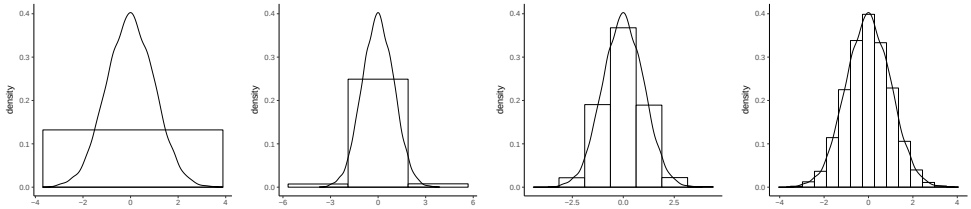
where the l_y bins $b_1, \ldots, b_{l_y}$ cover the space where $Y(t)$ takes its values. Then, to estimate $\bar{q}_{0,w}$, we in fact learn each of

$$P_0^N\left(Y(t) \in \cup_{l' \leqslant l} b_{l'} \mid Y_c^-(t), Y(t) \notin \cup_{l' < l} b_{l'}\right)$$

(every $1 \leqslant l < l_y$) and estimate the conditional distribution of $Y(t)$ given $Y_c^-(t)$ and $Y(t) \in b_l$ with the uniform distribution over b_l (we will improve this in near future, by making the conditional distribution more data-adaptive, to reflect for instance the empirical conditional means and variances). We proceed likewise for $W(t)$, one coordinate at a time.

The choice of the number of bins, say l_y, influences directly the level of smoothing performed, smaller values corresponding to more smoothing. Figure 8.1 shows a schematic of the smoothing step and the effect of the number of bins on the smoothness of the density.

In Section 8.5.2, we discussed why one should not choose *a priori* one algorithm in a collection of *a priori* specified algorithms, but should instead select it based on data. The discussion is very relevant here too, focusing on the number of bins. Likewise, we do not know beforehand what is the best number of bins to choose and we should select it based on data. This can be formalized as follows in the framework of Section 8.5.2. Given a set $\mathcal{L}$ of candidate number of bins, each couple of individual algorithm Φ_k and $l \in \mathcal{L}$ yields a (k, l)-specific algorithm $\Phi_{k,l}$ corresponding to setting the number of bins to l prior to using Φ_k. We could then exploit CV to data-adaptively identify and select the best (k, l) in $\{1, \ldots, K\} \times \mathcal{L}$.



(a) $l = 1$ bin. **(b)** $l = 3$ bins. **(c)** $l = 7$ bins. **(d)** $l = 15$ bins.

Figure 8.1: Influence of the number of bins on the amount of smoothing.

Online cross-validation on a time series.

Because of the chronological ordering and sequential dependence at play in the time series we observe, standard CV like the V-fold instance introduced in Section 8.5.2 cannot be used out of the box. Instead, we use a form of CV described by Benkeser et al. (2016) that can handle sequentially dependent data.

In view of Equation (8.14) and for the negative $\log$-likelihood loss function ℓ, we define recursively the online cross-validated risk of an algorithm Φ devised to learn $\bar{q}_{0,w}$, $\bar{g}_0$, or $\bar{q}_{0,y}$ as follows: for a sample size $1 \ll n \ll N$ deemed sufficiently large to start learning from $\bar{O}(n)$ but not as large as N,

$$R_{\ell,n}(\Phi) = \ell\left(\Phi(\bar{O}(n))\right)(O(n+1))$$

and, for every $n < t < N$,

$$R_{\ell,t}(\Phi) = \frac{t-n}{t+1-n} R_{\ell,t-1}(\Phi) + \frac{\ell\left(\Phi(\bar{O}(t))\right)(O(t+1))}{t+1-n}, \tag{8.22}$$

where $\Phi(\bar{O}(t-1)$ is the element of $\mathcal{Q}_w$ or $\mathcal{G}$ or $\mathcal{Q}_y$ learned by Φ based on $\bar{O}(t-1)$ and $\ell(\Phi(\bar{O}(t-1)))(O(t))$ equals Equation (8.17), Equation (8.18), or Equation (8.19) (without the subscript j) depending on the feature that is being learned by Φ. In words, after algorithm Φ has been trained on an initial data set of size n, one recursively uses $\bar{O}(t)$ to train Φ and $O(t+1)$ to determine how well the resulting $\Phi(\bar{O}(t))$ performs on the unseen $O(t+1)$, the overall measure of performance at sample size t being the average across the performances at sample sizes between n and t of the individual measures.

If Φ is an *online learning* machine learning algorithm, i.e., if the derivation of $\Phi(\bar{O}(t+1))$ recursively consists in updating the previous $\Phi(\bar{O}(t))$, then the sequential computation of $R_{\ell,t}(\Phi)$ is online too. This fact is of paramount importance, as it circumvents the occurrence of what would be otherwise a computational bottleneck.

Online SuperLearner algorithm.

We adapt the concepts of ensemble learning and online learning for the machine learning of conditional densities such as $\bar{q}_{0,w}$, $\bar{g}_0$, and $\bar{q}_{0,y}$ by combining what we have presented so far in the present section. The resulting Online SuperLearner (OSL) capitalizes on the classical SuperLearner algorithm. As its name suggests, it is online, scales to large data sets and, therefore, can deal with streaming and time series data by constantly updating the learner.

A high-level pseudo code and description of a generic algorithm for the estimation of a conditional density as discussed above are presented in Appendix D.2.1. Another high-level pseudo code and description of the sampling procedure based on discretization as discussed above are given in Appendix D.2.2.

A flowchart that describes the general OSL procedures is shown in Figure 8.2. Two comments are in order. First, the OSL does not rely on all data being available when we train the estimators. Once sufficiently many blocks of data have been observed to start learning from them, the estimator at sample size $(t+1)$ is derived by updating the estimator built at time t based on using the $(t+1)^{\text{th}}$ block (and not by retraining the algorithm on the whole data set). Second, in order for the OSL to be a truly online algorithm, each of its candidate algorithms must be online algorithms as well (Benkeser et al., 2016). If any candidate learner is not an online algorithm (viz., a batch algorithm; Hastie et al., 2009), it would require all of the data to be available upon training, and would therefore defeat the purpose of an OSL.

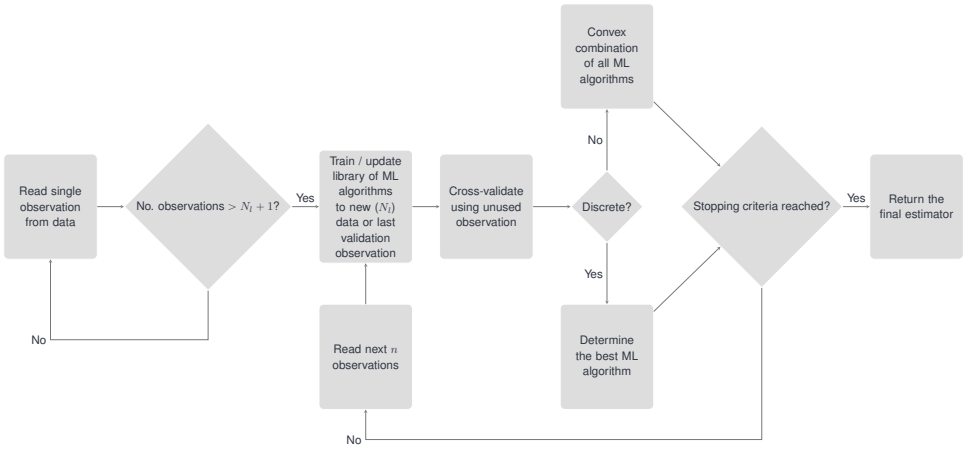


Figure 8.2: Schematic representation of the OSL procedure.

8.5.4 Step two: targeting the parameters of interest

Set arbitrarily $1 \leqslant j \leqslant J$. Let $\bar{q}_w^N, \bar{g}^N, \bar{q}_y^N$ be the estimators of $\bar{q}_{0,w}, \bar{g}_0, \bar{q}_{0,y}$ produced by OSL based on O_j^N as described in Section 8.5.3. For convenience, let η_j^N be equal to $(\bar{q}_w^N, \bar{g}^N, \bar{q}_y^N)$. In this section, we denote $P^N[\eta_j^N]$ the element of model $\mathcal{M}$ that is uniquely identified by η_j^N. For every $a \in \mathcal{A}$ and $1 \leqslant s < N$, $\psi_{j,s,a}^N(P^N[\eta_j^N])$ estimates $\psi_{j,s,a}^N(P_0^N)$. Recall that Section 8.4.3 discusses three assumptions under which $\psi_{j,s,a}^N(P_0^N)$ equals $\mathrm{CQ}_{0,j,s,a}^N$, the effect in individual j of imposing activity a at time s as measured in terms of average outcome at time $(s+1)$, see Section 8.3.3.

Monte-carlo procedure.

From a mathematical point of view, $\psi_{j,s,a}^N(P^N[\eta_j^N])$ is an integral. Its value cannot be derived analytically (van der Laan & Rose, 2017). We thus rely on the *Monte-Carlo* procedure to approximate it.

The Monte-Carlo procedure that we apply unfolds as follows. A large number B of times, we sequentially draw an observation $\bar{O}^b(s+1)$ from P_j^N and store the corresponding outcome at time $(s+1)$, namely $Y^b(s+1)$. Finally, the average $B^{-1}\sum_{b=1}^B Y^b(s+1)$ approximates $\psi_{j,s,a}^N(P^N[\eta_j^N])$. More specifically, omitting the b-superscript for clarity, starting from the observed $W_{j,c}^-(1) = \gamma_{w,1}(W_j^-(1))$, we draw $W(1)$ from the conditional distribution $\bar{q}_w^N$ given $W_{j,c}^-(1)$; we evaluate $A_{j,c}^-(1)$, draw $A(1)$ from the conditional distribution $\bar{g}^N$ given $A_{j,c}^-(1)$ if $s \neq 1$ or impose $A(1) = a$ otherwise; we evaluate $Y_{j,c}^-(1)$, draw $Y(1)$ from the conditional distribution $\bar{q}_y^N$ given $Y_{j,c}^-(1)$; then, recursively, as long as $t \leqslant s+1$, we draw $W(t)$ from the conditional distribution $\bar{q}_w^N$ given $W_{j,c}^-(t)$; we evaluate $A_{j,c}^-(t)$, draw $A(t)$ from the conditional distribution $\bar{g}^N$ given $A_{j,c}^-(t)$ if $s \neq t$ or impose $A(t) = a$ otherwise; we evaluate $Y_{j,c}^-(t)$, and draw $Y(t)$ from the conditional distribution $\bar{q}_y^N$ given $Y_{j,c}^-(t)$. A pseudo code example of this procedure is given in Appendix D.2.3. The algorithm used for generally sampling from a conditional distribution is provided in Appendix D.2.2.

The online one-step estimator.

For reasons already discussed in Section 8.5.1, the estimator $\psi_{j,s,a}^N(P^N[\eta_j^N])$ does not lend itself well to the construction of a confidence interval. However, it can be used to build a better-behaved estimator coined 'online one-step estimator' because (i) it is defined online, and (ii) a so called one-step correction is applied. Explaining the theory that leads to the definition of $\psi_{j,s,a}^N(P^N[\eta_j^N])$ is beyond the scope of this chapter. We refer the interested reader to Theorem 19.3 in (van der Laan & Rose, 2017) for details. This said, we nevertheless wish to define $\psi_{j,s,a}^N(P^N[\eta_j^N])$ properly.

Assume for simplicity that we can derive η_j^t starting from $t = 1$. Fix arbitrarily $2 \leqslant t \leqslant N$. Let $\pi\bar{O}_j(t)$ be block $O_j(t)$ augmented with its relevant history, namely

$$\pi\bar{O}_j(t) = (W_{j,c}^-(t), W_j(t), A_{j,c}^-(t), A_j(t), Y_{j,c}^-(t), Y_j(t)).$$

Moreover, let $h_w[\eta_j^{t-1}]$, $h_a[\eta_j^{t-1}]$ and $h_y[\eta_j^{t-1}]$ be the marginal densities of $W_{j,c}^-(t)$, $A_{j,c}^-(t)$ and $Y_{j,c}^-(t)$ under $P^N[\eta_j^{t-1}]$. Likewise, let $h_{w(t)}^*[\eta_j^{t-1}]$, $h_{a(t)}^*[\eta_j^{t-1}]$ and $h_{y(t)}^*[\eta_j^{t-1}]$ be the marginal densities of $W_{j,c}^-(t)$, $A_{j,c}^-(t)$ and $Y_{j,c}^-(t)$ under the unique element of $\mathcal{M}$ identified by η_j^{t-1} *but* such that $A_j(s)$ is set to a, see Equation (8.8). Recall Equation (8.10) and define in the same spirit

$$Z_{s,a}^{t-1} = \frac{\mathbb{I}\{A(s) = a\}}{\bar{g}^{t-1}(A(s) \mid A_c^-(s))}.$$

Finally, let $D_{j,s,a}[\eta_j^{t-1}](\pi\bar{O}_j(t))$ be given by

$$\begin{aligned} D_{j,s,a}[\eta_j^{t-1}](\pi\bar{O}_j(t)) = \sum_{r=1}^{s+1} \Big(& D_{j,s,a,1}^r[\eta_j^{t-1}](W_j(t), W_{j,c}^-(t)) \\ & + D_{j,s,a,2}^r[\eta_j^{t-1}](A_j(t), A_{j,c}^-(t)) \\ & + D_{j,s,a,3}^r[\eta_j^{t-1}](Y_j(t), Y_{c,j}^-(t)) \Big) \end{aligned}$$

with

$$\begin{aligned} D_{j,s,a,1}^r[\eta_j^{t-1}](W_j(t), W_{j,c}^-(t)) = & \frac{h_{w(r)}^*[\eta_j^{t-1}](W_{j,c}^-(t))}{h_w[\eta_j^{t-1}](W_{j,c}^-(t))} \\ & \times \Big(\mathbb{E}_{P^N[\eta_j^{t-1}]}[Y(s+1) \times Z_{s,a}^{t-1} \mid W(s) = W_j(t), W_c^-(s) = W_{j,c}^-(t)] \\ & - \mathbb{E}_{P^N[\eta_j^{t-1}]}[Y(s+1) \times Z_{s,a}^{t-1} \mid W_c^-(s) = W_{j,c}^-(t)] \Big), \end{aligned}$$

$$\begin{aligned} D_{j,s,a,2}^r[\eta_j^{t-1}](A_j(t), A_{j,c}^-(t)) = & \mathbb{I}\{r \neq s\} \frac{h_{a(r)}^*[\eta_j^{t-1}](A_{j,c}^-(t))}{h_a[\eta_j^{t-1}](A_{j,c}^-(t))} \\ & \times \Big(\mathbb{E}_{P^N[\eta_j^{t-1}]}[Y(s+1) \times Z_{s,a}^{t-1} \mid A(s) = A_j(t), A_c^-(s) = A_{j,c}^-(t)] \\ & - \mathbb{E}_{P^N[\eta_j^{t-1}]}[Y(s+1) \times Z_{s,a}^{t-1} \mid A_c^-(s) = A_{j,c}^-(t)] \Big), \quad \text{and} \end{aligned}$$

$$\begin{aligned} D_{j,s,a,3}^r[\eta_j^{t-1}](Y_j(t), Y_{j,c}^-(t)) = & \frac{h_{y(r)}^*[\eta_j^{t-1}](Y_{j,c}^-(t))}{h_y[\eta_j^{t-1}](Y_{j,c}^-(t))} \\ & \times \Big(\mathbb{E}_{P^N[\eta_j^{t-1}]}[Y(s+1) \times Z_{s,a}^{t-1} \mid Y(s) = Y_j(t), Y_c^-(s) = Y_{j,c}^-(t)] \\ & - \mathbb{E}_{P^N[\eta_j^{t-1}]}[Y(s+1) \times Z_{s,a}^{t-1} \mid Y_c^-(s) = Y_{j,c}^-(t)] \Big). \end{aligned}$$

Then the online one-step estimator (OOS) ψ_N^{oos} can be expressed as

$$\psi_N^{\text{oos}} = \frac{1}{N-1} \sum_{t=2}^{N} \Big(\psi_{j,s,a}^N(P^N[\eta_j^{t-1}]) + D_{j,s,a}[\eta_j^{t-1}](\pi \bar{O}_j(t)) \Big). \tag{8.23}$$

Monte-Carlo approximation of the correction term.

Consider Equation (8.23). In Section 8.5.4, we have already explained how we approximate $\psi_{j,s,a}^N(P^N[\eta_j^{t-1}])$ by Monte-Carlo. To fully operationalize the computation of ψ_N^{oos}, it remains to explain how we derive $D_{j,s,a}$, the difficulty lying in approximating the ratios $h^*_{w(r)}[\eta_j^{t-1}]/h_w[\eta_j^{t-1}]$, $h^*_{a(r)}[\eta_j^{t-1}]/h_a[\eta_j^{t-1}]$, and $h^*_{y(r)}[\eta_j^{t-1}]/h_y[\eta_j^{t-1}]$ on the one hand; and the various conditional expectations that appear in the definitions of $D^r_{j,s,a,1}$, $D^r_{j,s,a,2}$ and $D^r_{j,s,a,3}$ ($1 \leqslant r \leqslant s+1$), on the other hand.

For the various conditional expectations, we rely again on the Monte-Carlo procedure. When addressing the approximation of, say,

$$\mathbb{E}_{P^N[\eta_j^{t-1}]}[Y(s+1) \times Z_{s,a}^{t-1} \mid W(s) = W_j(t), W_c^-(s) = W_{j,c}^-(t)],$$

the approximation goes along the same lines as that of $\psi_{j,s,a}^N(P^N[\eta_j^{t-1}])$, except that we start from $(W(s), W_c^-(s)) = (W_j(t), W_{j,c}^-(t))$.

For the ratios, we use a trick that frames the task as a classification problem. Consider for concreteness the ratio $h^*_{y(r)}[\eta_j^{t-1}]/h_y[\eta_j^{t-1}]$. Let B be a large integer. Let $T_1, \ldots, T_B$, $O^1, \ldots, O^B$, and $O^{*1}, \ldots, O^{*B}$ be $3 \times B$ random variables independently drawn from the uniform distribution on $\{1, \ldots, N\}$ (T_b, $1 \leqslant b \leqslant B$), $P^N[\eta_j^{t-1}]$ (O^b, $1 \leqslant b \leqslant B$), and from the unique element of $\mathcal{M}$ identified by η_j^{t-1} *but* such that $A_j(s)$ is set to a, see Equation (8.8) (O^{*b}, $1 \leqslant b \leqslant B$). For every $1 \leqslant b \leqslant B$, let

$$\Gamma_b = \gamma_{y,T_b}(Y^-(T_b))$$

where $Y^-(T_b)$ is the past of $Y(T_b)$ in O^b. Likewise, let

$$\Gamma_b^* = \gamma_{y,r}(Y^-(r))$$

where $Y^-(r)$ is the past of $Y(r)$ in O^{*b}. Now, let $(\bar{\Gamma}_1, \Delta_1), \ldots, (\bar{\Gamma}_{2B}, \Delta_{2B})$ be defined in such a way that

$$\{\bar{\Gamma}_1, \ldots, \bar{\Gamma}_{2B}\} = \{\Gamma_1, \ldots, \Gamma_B, \Gamma_1^*, \ldots, \Gamma_B^*\}$$

and $\Delta_b = 1$ if $\bar{\Gamma}_b \in \{\Gamma_1, \ldots, \Gamma_B\}$ and $\Delta_b = 0$ otherwise.

We view $(\bar{\Gamma}_1, \Delta_1), \ldots, (\bar{\Gamma}_{2B}, \Delta_{2B})$ as independent random variables drawn from the law Π of $(\bar{\Gamma}, \Delta)$ such that $\Delta = 1$ with probability $\frac{1}{2}$ and, given Δ, $\bar{\Gamma}$ is drawn from

$h_y[\eta_j^{t-1}]$ if $\Delta = 1$ and from $h^*_{y(r)}[\eta_j^{t-1}]$ otherwise. Let $H = (h_y[\eta_j^{t-1}] + h^*_{y(r)}[\eta_j^{t-1}])/2$ be the marginal density of $\bar{\Gamma}$ under Π. By Bayes' rule, it holds that

$$\begin{aligned}\Pi(\Delta = 1 \mid \bar{\Gamma}) &= h_y[\eta_j^{t-1}](\bar{\Gamma})/2H(\bar{\Gamma}),\\ \Pi(\Delta = 0 \mid \bar{\Gamma}) &= h^*_{y(r)}[\eta_j^{t-1}](\bar{\Gamma})/2H(\bar{\Gamma}),\end{aligned}$$

hence

$$\frac{h^*_{y(r)}[\eta_j^{t-1}](\bar{\Gamma})}{h_y[\eta_j^{t-1}](\bar{\Gamma})} = \frac{1 - \Pi(\Delta = 1 \mid \bar{\Gamma})}{\Pi(\Delta = 1 \mid \bar{\Gamma})},$$

revealing that the ratio can be approximated by carrying out the machine learning of the conditional probability of $\Delta = 1$ given $\bar{\Gamma}$.

8.6 Simulation study

We performed a preliminary simulation study to evaluate the performance of our implementation of the OSL and the OOS. In a simulation study one tests the performance of an estimator using synthetically crafted data with known ground truth, constructed to mimics nature. Because the ground truth of the data generating process is known (or well approximated), one can reliably evaluate the performance of the estimators of this process by calculating the difference between the estimate and the truth. We evaluate the approaches described in Section 8.5 using this simulation study.

8.6.1 Simulation scheme

For our simulation scheme we introduce the set $\mathcal{S}$ of data-generating distributions for O^N satisfying the following description: a certain distribution $P_1^N \in \mathcal{S}$ if there exist a shared in time m_w-dimensional parameter $\theta_w \in \mathbb{R}^{m_w}$, a shared in time m_a-dimensional parameter $\theta_a \in \mathbb{R}^{m_a}$, a parameter $\delta \in (0, \frac{1}{2})$, and a real-valued function μ over $\{0,1\} \times \mathbb{R}$ such that an observation O^N drawn from the distribution P_1^N can be derived using the following procedure:

1. Draw a random sample from $U_W(t)$ for $t \in \{1 - \max(m_w, m_a), \ldots, N\}$ independently from the standard normal distribution $\mathcal{N}(0,1)$.
2. Conditionally on this random sample drawn from $U_W(t)$, draw a random sample $U_A(t)$ from the Bernoulli distribution with parameter $\mathrm{expit}(U_W(t))$ for every $t \in \{1 - \max(m_x, m_a), \ldots, N\}$. The expit function is used to bound the outcome between zero and one.

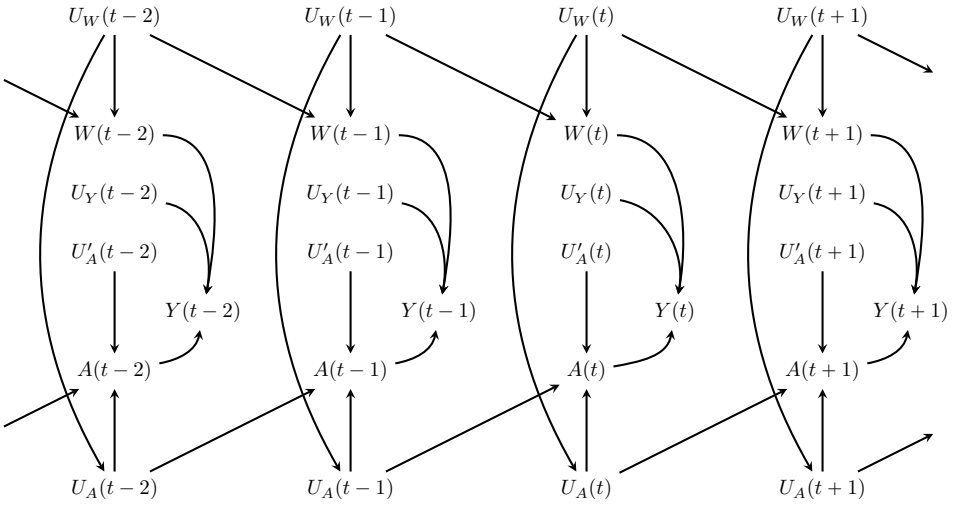


Figure 8.3: Directed acyclic graph representing the dependence structure of an element P_1 of model $\mathcal{S}$. One reads in the directed acyclic graph (DAG) that the parameters θ_w and θ_a attached to P_1 are both two-dimensional.

3. Define $W(t) = \sum_{s=1}^{m_w} \theta_w(s) U_W(t+1-s)$ for every $t \in \{1, \ldots, N\}$.

4. Conditionally on the random variables drawn or defined so far, sample $A(t)$ from the Bernoulli distribution with parameter

$$\delta + (1-2\delta)\,\mathrm{expit}\left(\sum_{s=1}^{m_a} \theta_a(s) U_A(t+1-s)\right),$$

for every $t \in \{1, \ldots, N\}$.

5. Conditionally on the random variables drawn or defined so far, sample $Y(t)$ from the Normal distribution $\mathcal{N}(\mu(A(t), W(t)), 1)$ for every $t \in \{1, \ldots, N\}$.

6. Finally, define a single observation $O^N = (O(1), \ldots, O(N))$ with $O(t) = (W(t), A(t), Y(t))$ for every $t \in \{1, \ldots, N\}$.

The simulation structure can be structured as a DAG to present the dependency structure. Figure 8.3 presents the dependence structure of an element of $\mathcal{S}$, say P_1^N. Inspecting the directed acyclic graph notably reveals that, under P_1^N, the conditional distribution of $W(t)$ given its past $W^-(t)$ coincides with its conditional distribution given $O(t-1)$; the conditional distribution of $A(t)$ given its past $A^-(t)$ coincides with its conditional distribution given $(O(t-1), W(t))$; the conditional distribution

of $Y(t)$ given its past $Y^-(t)$ coincides with its conditional distribution given $(O(t-1), W(t), A(t))$.

8.6.2 Implementation

The OSL and the OOS are both implemented in the R-statistical environment as an open source R-package (R Development Core Team, 2008). Besides the OSL and the OOS, the package also contains the R-based implementation of the simulator[12].

The created simulator can be used as follows. The R-implementation of the simulator exposes a function `SimulateWAY` that accepts a number of arguments, the most important ones being: `numberOfBlocks`, `qw`, `ga`, `qy`, and `intervention`. The first argument, `numberOfBlocks`, defines the number of blocks one would like to simulate. The arguments `qw`, `ga`, and `qy` define the *true* conditional distributions $q_{0,\bar{q}_w}(\cdot)$, $g_{0,\bar{g}}(\cdot)$, and $q_{0,\bar{q}_y}(\cdot)$. For each of these conditional distributions, one has to define on how many previous measurements each of these variables depend (the 'memory' of the data generating process). Furthermore, each of these processes should define a stochastic mechanism, that is, the random mechanism that is combined with the memory of a mechanism and is used as input for the data generating function. Lastly one can supply an intervention using the `intervention` parameter. An intervention is specified as a list of three vectors, one vector specifying which `variable` is currently under intervention, one vector representing `when` the intervention should take place (i.e., at which time t in the data generating process) and the other vector `what` the intervention should be (i.e., an activity $a \in \mathcal{A}$). Specifying this intervention for the simulation scheme allows us to sample our data from any distribution in $\mathbb{P}_0^N$. The product of this simulator is a matrix of blocks, with $N =$ `numberOfBlocks` rows, and $\|W\|_0 + 2$ columns.

The implementation of the OSL exposes a number of functions, of which the `initialize`, `fit`, `predict`, and `sampleIteratively` functions of the `OnlineSuperLearner` class are of particular interest. The initialize function is used to create a new OSL instance. The function expects (among others arguments) a list of candidate algorithms and options describing which types of OSL to fit (discrete, the convex combination, or both). After initialization, the `fit` function can be used to learn the conditional densities. This can be done by providing the `fit` function with either a previously collected data source, or by providing a streaming data source. After learning the conditional densities, both the `predict` and `sampleIt-`

[1]The source code for the simulator and the OOS are available on `https://github.com/frbl/onlinesuperlearner`.

[2]Note that this package is still under active development, and currently supports learning the conditional densities and partly supports performing the one-step correction using the OOS. Estimation of confidence intervals is currently not yet supported and will be addressed in detail in future work.

`eratively` functions can be used to access the learned conditional densities, either to predict probabilities, or to sample new values from them.

8.6.3 Simulation results

We evaluated the performance of our implementations of the OSL and the OOS using four preliminary simulation configurations, each conveying different data generating distributions. In the first two simulation configurations, we simulate a data set in which no time dependence is present ($Y(t)$ is only influenced by $A(t)$ and $W(t)$). In the third and fourth simulation configuration, we do include a time dependence, that is, $Y(t)$ depends on both $A(t)$ and $W(t)$, and on m previous observations $O(t-1)\dots O(t-m)$. In the first and third configuration, we assume $Y(t)$ is a random variable with a binomial distribution. In the second and fourth configuration, $Y(t)$ is considered to be continuous.

For each configuration, we specified an intervention with a binary treatment variable. We evaluated both treatment ($a = 1$) and control ($a = 0$) at $A(s)$ where $s = 2$. We measured the outcome at s for *Cfg 1* and *Cfg 2*, and at $s + 1$ for configurations *Cfg 3* and *Cfg 4* (because of the time dependency in the latter configurations). The simulator provides us with the true data generating function, and from that we can approximate the true $\text{CQ}^N_{0,j,s,a}$ by sampling a large number of B observations and averaging the outcome at time s or $s + 1$. For this simulation we used $B = 100$ iterations for each of the configurations and for the approximation of the truth.

For learning the conditional densities, we used observations of length $N = 1\,000$ blocks. We split this training set into 101 smaller parts and performed an initial training on $n = 500$ blocks. This initial training is used to partially train the algorithms before performing the update steps. The remaining $n = 500$ blocks were split up in 100 mini-batches, each containing five blocks. Although splitting the training data into several separate sets is not strictly necessary, it is needed for evaluating the online behavior of the estimators and calculating a reliable CV risk. For this initial simulation, we included five candidate estimators based on two algorithms: the XGBoost algorithm (a scalable machine learning system for gradient boosting; T. Chen & Guestrin, 2016), and a generalized linear model (based on the speedglm R-package Enea, Meiri, & Kalimi, 2017). For XGBoost, we included four different configurations of the α-parameter ($l1$ regularization parameter): zero and three values sampled from $\mathcal{U}(0, 1)$. For each estimator, we data-adaptively selected the best number of bins l used in the discretization step, which we restricted to $l \in \mathcal{L} = \{40, 50, 60, 70\}$. The best performing estimator was selected using the sequential CV procedure in Equation (8.22).

We trained and optimized each of these algorithms to predict the subsequent

measurement in time using the CV procedure provided in Equation (8.22). Then, after each algorithm had yielded a number of estimators, we used both the discrete and continuous OSL to form an initial estimation of our $\text{CQ}^N_{0,j,s,a}$. Finally, we used OOS to apply the one-step correction, yielding a corrected version of this quantity. For each iteration in the efficient influence curve we sampled 50 blocks, and calculated the correction term based on the first ten blocks in the time series. An overview of the cross-validated risk of each individual estimator is shown in Figure 8.4. These figures show the total CV risk of the estimators for each of the different simulation configurations.

Finally, we present the performance of the discrete OSL and the continuous OSL in Table 8.2. This table shows both the approximated ('true') quantity of interest and the estimated quantity as calculated by the OSL. We show this estimate both before and after applying the OOS update. Furthermore, to show the convergence of the continuous OSL and discrete OSL to the quantity of interest, we show eight convergence plots in Figure 8.5. This figure shows the number of bootstrap iterations (on the x-axis) and the value the estimators and the simulator are approximating (the y-axis). The top row of Figure 8.5 shows the convergence when we impose an intervention ($a = 1$), and the bottom row shows the convergence when control ($a = 0$) is imposed.

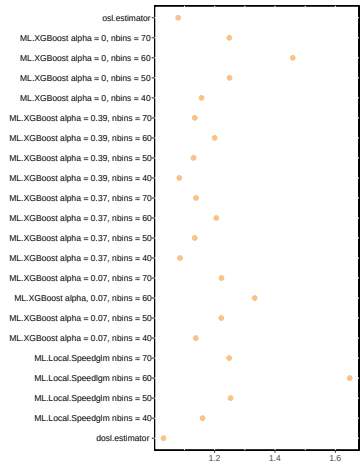
Table 8.2: Estimates for the Online SuperLearner and the Discrete Online SuperLearner for both the control and intervention data.

Configuration	Truth	OSL (pre OOS)	OSL (post OOS)	Discrete OSL (pre OOS)	Discrete OSL (post OOS)
Cfg 1 (intervention)	0.63	0.63	0.63	0.62	0.55
Cfg 1 (control)	0.40	0.45	0.65	0.44	0.57
Cfg 2 (intervention)	43.00	41.95	N/A	42.85	45.12
Cfg 2 (control)	43.02	42.10	N/A	42.04	47.98
Cfg 3 (intervention)	0.51	0.54	0.38	0.45	0.63
Cfg 3 (control)	0.57	0.55	0.22	0.62	0.71
Cfg 4 (intervention)	42.49	42.29	N/A	42.57	42.21
Cfg 4 (control)	42.43	42.27	N/A	42.98	40.00

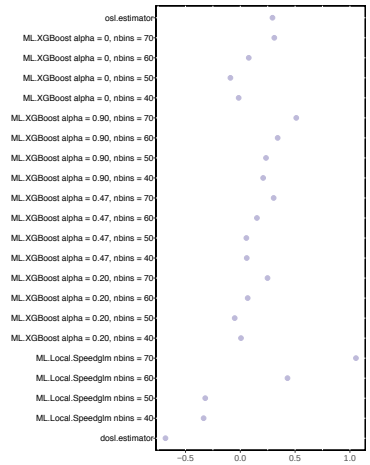
Note: The scores of the OSL (post OOS) failed to finish in certain cases. These cases have been marked with a N/A. Also, intermediate results of the failed OSL (post OOS) configurations showed that the estimate was diverging from, rather than converging to the truth.

8.7 Application to the HowNutsAreTheDutch data set

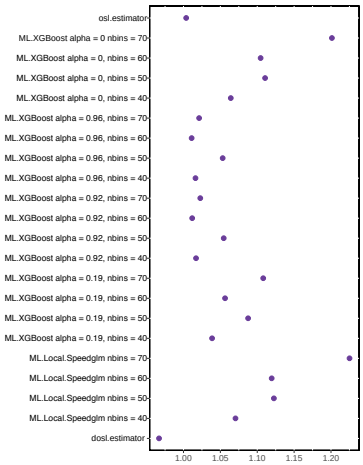
We used our implementation of the OSL for a preliminary analysis in the HND data set. For this data set, we aim to predict the effect of an activity $a \in \mathcal{A}$ on the level of positive affect one experiences. The used data set was exported from HND on February 19, 2015. We used data from all individuals that participated in the study up to that date, and that had completed more than 75% ($t = 68$) measurements. We used Amelia-II imputation for missing data. Amelia-II is a well validated method to



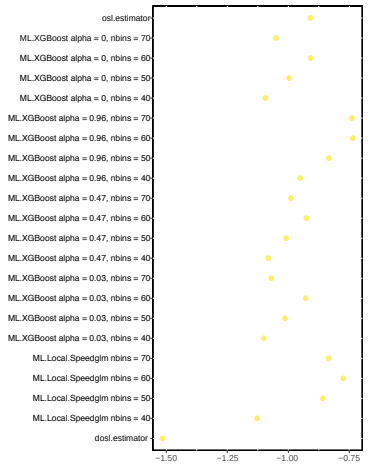
(a) Cross-validated risk on Cfg 1.



(b) Cross-validated risk on Cfg 2.



(c) Cross-validated risk on Cfg 3.

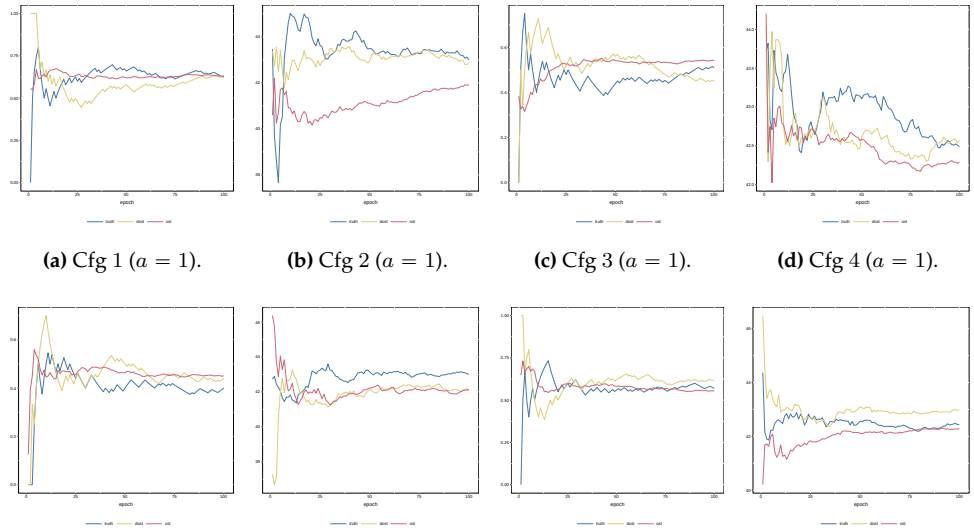


(d) Cross-validated risk on Cfg 4.

Figure 8.4: Graph showing the different estimators and their total cross-validated negative log-likelihood loss on one of the simulation configurations.

deal with missing data in a time series (Honaker & King, 2010; Honaker et al., 2011). In order to distinguish the effect per activity, we separated the categories using one-hot encoding[3]. We used the same algorithms as used in Section 8.6.3, with the same

[3]One-hot encoding is a method of recoding discrete categorical variables to a series of binary dummy variables, where each dummy represents one of the activities.



(a) Cfg 1 ($a = 1$). **(b)** Cfg 2 ($a = 1$). **(c)** Cfg 3 ($a = 1$). **(d)** Cfg 4 ($a = 1$).

(e) Cfg 1 ($a = 0$). **(f)** Cfg 2 ($a = 0$). **(g)** Cfg 3 ($a = 0$). **(h)** Cfg 4 ($a = 0$).

Figure 8.5: Convergence for each of the simulation configurations under treatment (row one) and control (row two). The blue line depicts the truth, the yellow line the discrete OSL, and the red line the continuous OSL.

configurations. The group level results of these analysis are provided in Table 8.3. As can be seen in Table 8.2, the current implementation of the OOS does not seem to improve the initial estimate. As such, we only applied the OSL to provide an estimate of the target quantities.

The outcomes presented in Table 8.2 show some interesting properties. Firstly, both the OSL and discrete OSL seem to be converge to similar outcomes. The point-estimates of the target quantity seems to differ only slightly for both estimators. The largest difference between the OSL and the discrete OSL is the difference for the tenth activity ('Web surfing / gaming / social media'), with an absolute difference of 3.64. Secondly, the estimates themselves. There do not seem to be large, meaningful differences in positive affect (PA) when intervening on the different activities at the preceding time, as all estimates seem to be relatively similar. In future versions of the OSL significance testing could reveal whether or not these differences are in fact statistically significant.

	Activity	OSL	Discrete OSL
1	Resting / sleeping	58.91	56.99
2	Household / groceries	59.38	59.59
3	Working / studying / volunteering	57.36	57.34
4	Exercising / walking / cycling	57.10	55.08
5	Yoga / meditation / sauna visit etc.	57.90	58.69
6	Reading	57.75	57.18
7	Hobby (e.g., gardening, making music)	53.05	55.90
8	Trip (e.g., leisure park, concert)	54.48	54.72
9	Watching TV	58.41	57.72
10	Web surfing / gaming / social media	54.35	57.99
11	Conversing	57.15	58.72
12	Something intimate (e.g., cuddling, sex)	59.99	59.65

Table 8.3: Results for each of the available activities.

8.8 Discussion and Concluding Remarks

We presented our implementation of the OSL. We provided its mathematical foundations and provided argumentation for the applied SuperLearner approach. We performed a simulation study to determine the performance of the OSL on four simulated time series datasets and showed that it is a viable method for time-series analysis. Moreover, we demonstrated the usefulness of the OSL approach to a real world psychopathology data set from HND. The estimations performed in both the simulation study and the HND example were performed with a reasonably small number of estimators and bootstrap iterations, and as such, the corresponding results should be considered preliminary. Furthermore, no elaborate hyperparameter tuning was performed to optimize the estimators. By selecting both a larger number and wider variety of machine learning-based estimators, and by performing a hyperparameters tuning and optimization step for each of the algorithms, one could further improve the quality of the resulting OSL instances.

Three evident directions for future work on the OSL and the OOS are (i) the improvement of the OOS estimates, (ii) the extension of the OSL with confidence intervals, (iii) the extension of the number of summary measures, and (iv) the implementation of different machine learning algorithms. Firstly, the improvement of the OOS estimates. Currently the OOS' performance is not optimal. Although it sometimes improves the initial estimate of the OSL, in some cases it actually hurts the estimates. More research should be performed to investigate what underlies this phenomenon.

Secondly, the estimation of confidence intervals. After the OOS implementation has been improved, and it has been shown to result in well-behaved estimators (i.e., estimators that converge to the truth, and are Gaussian), the next step is to introduce the estimation of confidence intervals (CIs). These CIs can provide insight into the differences between the estimates (e.g., in the HND example) and could provide support for significance testing.

Thirdly, the creation of new summary measures. Although the OSL currently supports the generation of lagged features, other features (e.g., a running mean or variance, or interactions) could also be informative and potentially improve the prediction quality of the OSL.

Fourthly, the addition of different machine learning algorithms. Recently, researchers have shown a renewed interest in the use of *artificial neural networks*, in particular the interest in *deep-learning* has increased greatly (Jones, 2014; LeCun et al., 2015). Deep-learning is a particular form of machine learning based on the neural connections in the human brain. The results of such deep-learning implementations have been ground-breaking, and could be a great asset to add to the OSL. By the established oracle inequality result, we can include one or a set of deep-learning algorithms in the OSL, possibly improving the initial estimates of the deep-learning based estimators. Other machine learning algorithms could also form an interesting avenue for the extension of OSL. Implementations like the CARET package could offer a number of different implementations to do so (Kuhn, 2008).

In terms of software, another direction for future work concerns the underlying architecture of OSL. The current implementation serves mainly as a proof-of-concept, and was created in the R-statistical environment. The R-statistical environment is a well known and widely used language among researchers (R Development Core Team, 2008). Although this implementation works well in a research setting, it might not be the best option when running the OSL in production on streams of high resolution data. Other big-data platforms such as Spark (Zaharia et al., 2016), Storm (Toshniwal et al., 2014), and Apache Mahout (The Apache Software Foundation, 2016) offer elaborate features to deal with these kinds of data, and an implementation on one of these platforms might improve scalability and reliability of the current approach.

Lastly, we envision large scale implementations of the OSL for providing personalized and automated feedback on a large scale. Applications such as HND already have basic and more elaborate techniques in place to provide such personalized feedback, however, non of these techniques make use of data-driven and semi-parametric statistics, let alone targeted estimation. The implementation of such techniques on a large scale could increase empowerment and provide tools for individuals to help improve their well-being.